

PATENT ABSTRACTS OF JAPAN

10002681-02

Cited Reference 6

(11)Publication number : 2000-047902

(43)Date of publication of application : 18.02.2000

(51)Int.Cl.

G06F 11/28

G06F 11/34

(21)Application number : 10-210671

(71)Applicant : HITACHI LTD

HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 27.07.1998

(72)Inventor : SHIMURA MITSUHIRO

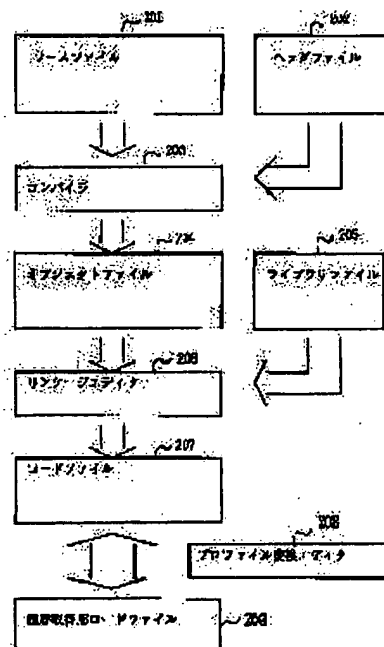
OBARA ISAO

(54) METHOD FOR CONTROLLING PROFILING

(57)Abstract:

PROBLEM TO BE SOLVED: To make the performance analysis efficient by selecting the specified function of an operation load program, converting it into a load program for sampling an execution history or inversely, converting it without sampling the execution history of the converted program being converted.

SOLUTION: A compiler 203 outputs an object file 204 having a redundant instruction with a source file 201 and a header file 202 as an input. A linkage editor 206 output a load file 207 including address information to the redundant instruction with the object file 204 and a library file 205 being its aggregation body as an input. The load file 207 is converted into a history information obtaining load file 209 by selecting the specified function by a profile conversion editor 208 or, on the contrary, the history information obtaining load file 209 is converted into the normal load file 207 without the execution of a history sampling processing. Thus, a history sampling object function is successively narrowed so that the execution time is shortened.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

10002681-02
Cited Reference 6

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-47902

(P2000-47902A)

(43) 公開日 平成12年2月18日 (2000.2.18)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 11/28	3 1 0	G 0 6 F 11/28	3 1 0 B 5 B 0 4 2
11/34		11/34	S

審査請求 未請求 請求項の数1 OL (全 6 頁)

(21) 出願番号 特願平10-210671

(22) 出願日 平成10年7月27日 (1998.7.27)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式会社

神奈川県横浜市中区尾上町6丁目81番地

(72) 発明者 志村 光弘

神奈川県横浜市中区尾上町6丁目81番地

日立ソフトウェアエンジニアリング株式会社
社内

(74) 代理人 100068504

弁理士 小川 勝男

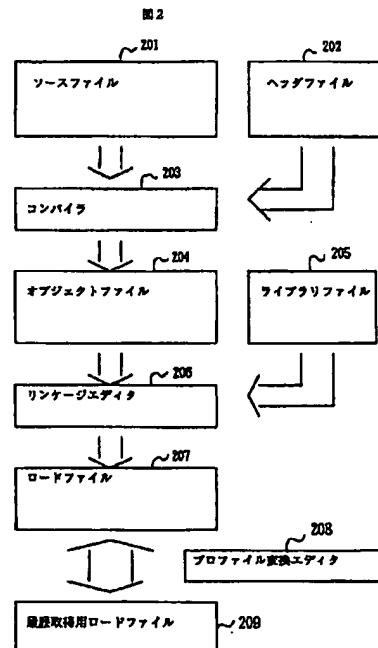
最終頁に続く

(54) 【発明の名称】 プロファイリング制御方法

(57) 【要約】

【課題】 計算機システムでソースプログラムの再コンパイルなしに、本番用のロードプログラムから実行履歴を採取するロードプログラムへ変換することを可能とするプロファイリング制御方法。

【解決手段】 ソースプログラムのコンパイル時のオブジェクトファイルに冗長命令を持たせて、そのオブジェクトファイルをリンケージしてロードプログラムを作成した時点で、その冗長命令へのアドレス情報をロードプログラムファイルに格納しておくことにより、特定関数の開始処理の冗長命令を、履歴採取関数への分岐命令に変更する。



【特許請求の範囲】

【請求項1】 計算機システムでのロードプログラムの実行時の履歴採取制御方法において、ソースプログラムのコンパイル時のオブジェクトファイルに冗長命令を持たせておくことと、そのオブジェクトファイルをリンケージしてロードプログラムを作成した時点で、その冗長命令へのアドレス情報をロードプログラムファイルに格納しておくことにより、ソースプログラムの再コンパイルなしに実行履歴を採取しないロードプログラムから全体、または特定の関数の実行履歴を採取するロードプログラムへ変換する手段と、実行履歴を採取するロードプログラムの全体または特定の関数の実行履歴を採取しないロードプログラムに変換する手段を有することを特徴とするプロファイリング制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ロードプログラムの実行時の履歴採取制御方法に関する。

【0002】 本制御方式を適用する業務は、大規模科学技術計算のためのプログラムのような実行履歴を採取するためにソースプログラムの再コンパイルするコストが大きく、実際にコンパイルを行うのが困難な業務が考えられる。

【0003】

【従来の技術】 「UNIX System V/386リリース4 プログラマーズ・リファレンス・マニュアル」ISBN4-8101-8515-X のprof(1) 1～ prof(1) 4 ページに示すように、プロファイリング（実行履歴）採取のためには、その為の特殊なオプション（-p）でソースプログラムをコンパイルし、オブジェクト中に実行履歴採取のための情報採取関数呼び出しを挿入した、プロファイリング専用のオブジェクトと、それをリンケージしたロードプログラムを再作成する方法がある。

【0004】 しかし、この方法では、実行履歴採取対象のプログラムのソースの規模が大きい場合、再コンパイルするためのコストが大きくなり、実際に再コンパイルを行うのは困難であることが多い。

【0005】 また、プログラムで使用するライブラリを他者から提供を受けていた場合にライブラリ自体のソースプログラムが手に入らず、履歴採取用のオブジェクトが作成できず、該当ライブラリ中に性能のボトルネックがあっても解析できない場合があった。

【0006】 また、「UNIXカーネルの設計」ISBN4-320-02551-2の225～227ページに示すように、クロック割り込みにてプログラムカウンタを採取することによりサンプルとして実行情報を採取できるが、この方法では関数のコール数や関数間の呼び出し関係までの履歴採取までは行えず、性能解析データとしての情報量が少ない。

【0007】

【発明が解決しようとする課題】 実際の性能解析に効果が高いプログラムの履歴情報を採取する為には、プログラム内で使用する各関数（手続き、プロシジャ、サブルーチンとも呼ばれるが、以下では関数と呼ぶ）の開始処理において、関数の呼び出し元（コール関係）を取得する履歴採取関数を呼び出す必要がある。また、本番運用時には、履歴採取関数の実行オーバーヘッドが大きいためこれを取り除ける仕掛けが必要である。しかも、全ての関数を常に履歴採取するのは、実行時間が大きくなるため、履歴採取の対象関数を、漸次絞っていくために、特定の関数のみ履歴採取としたり、逆に履歴を採取しなかったり、容易に行うことを可能とする必要がある。

【0008】 本発明の目的は、上記の要求を満たすような、実行履歴を採取するためにソースプログラムを再コンパイルすることなしに、本番運用用のロードプログラムの全部または特定の関数を選択して実行履歴採取用のロードプログラムに変換する手段と、逆に実行履歴採取用のロードプログラムの全部または特定の関数を実行履歴を採取しないように変換する手段を有する実行履歴採取制御方法を提供することにある。

【0009】

【課題を解決するための手段】 ソースプログラムのコンパイル時のオブジェクトファイルに関数のテキスト命令部の開始部分毎にあらかじめ実行時の性能に影響が小さいNOP（NO OPERATION）命令からなる冗長命令を展開し、その冗長命令へのオフセット情報をオブジェクトファイルに格納しておく。そして、そのオブジェクトファイルをリンケージしてロードプログラムを作成した時点で、その冗長命令へのアドレス情報をロードプログラムファイルにマージして格納しておくことにより、実行履歴が必要になった時点で、冗長命令へのアドレス情報より対象関数の冗長命令を位置付け、該当関数の冗長命令を実行履歴採取関数に分岐するように変更することにより実現する。

【0010】

【発明の実施の形態】 以下、図面を用いて本発明の一実施例を説明する。

【0011】 図1は、本発明のシステム構成例を示したものである。計算機システムは、中央処理装置101と磁気ディスク107から構成される。中央処理装置は、命令を実行する命令実行プロセッサ102と命令実行プロセッサ102が入出力操作なしにアクセス可能な主記憶装置103から構成される。コンパイラ104、リンケージエディタ105、プロファイル変換エディタ106は、磁気ディスク上のファイルと入出力を行う。

【0012】 図2は、本発明のファイル構成を示したものである。ソースファイル201とヘッダファイル202を入力として、コンパイラ203はオブジェクトファイル204を出力する。オブジェクトファイル204とオブジェクトファイルの集合体であるライブラリファイ

ル205を入力として、リンケージエディタ206は、ロードファイル207を出力する。このロードファイルを、プロファイル変換エディタ208により、履歴情報取得用ロードファイル209に変換したり、逆に履歴情報取得用ロードファイル209から履歴採取処理が実行されない通常のロードファイル207に変換することを行う。

【0013】図3は、本発明におけるオブジェクトファイルへのテキストの展開方式の一例を示したものである。func1というシンボル名称の関数を含むソースファイル301をコンパイルしてオブジェクトファイル302を作成する。この時、func1のテキストの開始処理に予め冗長なNOP命令を3命令埋め込み、オブジェクト中の冗長テキストコードアドレスポインタ格納領域303に、そのfunc1の開始アドレスから、そのNOP命令の開始アドレスへのオフセットを格納しておく。

【0014】図4は、ロードプログラムのリンケージ方式を示す。リンケージエディタは、オブジェクトファイル1(401)とオブジェクトファイル2(402)のオブジェクトを結合し(リンケージ)してロードプログラムを作成し、関数func1のリンケージ後のアドレスを確定後、オブジェクトファイル中に格納していた冗長コードアドレスへのオフセット情報を加算し、ロードプログラムファイルへ冗長テキストコードのリンケージ後のテキスト開始位置からのオフセットを、冗長テキストコードアドレスポインタ格納領域(以下プロファイルポインタ情報と呼ぶ)404に格納する。

【0015】図5は、履歴情報を採取しないロードファイルのテキスト501の冗長なNOP命令を、履歴情報取得のためのロードプログラムのテキストに変換している。変換後では、このNOP命令を引数レジスタ2にコール関係が判別できる情報を格納して、履歴情報採取関数に分岐させる命令に変換している。この変換するために、ロードプログラムのプロファイルポインタ情報502、504を参照し、変換後は、変換していることを識別するためのフラグをオンとして格納している。

【0016】図6は、ロードファイルのファイル形式を示す。ヘッダ構造体601には、このロードファイル中のロードプログラムのテキスト情報602、データ情報603、シンボル情報604、プロファイルポインタ情報604へのオフセットと各々のサイズの情報が格納される。

【0017】図7は、プロファイルポインタ情報605の詳細なファイル形式を詳細を示す。プロファイルポインタヘッダ構造体701は、このプロファイルポインタで管理する関数のシンボルの数が格納されている。ロードプログラムに格納されている関数1のシンボル名702、その関数内の冗長テキストコードアドレス702、その関数1の冗長コードが履歴取得用に変換されているか、またはNOP命令のまま変換されていないかをしめ

すフラグ703の3セットの情報関数毎に組みになり格納される。

【0018】図8は、リンケージ方式のフローチャートを示す。リンケージエディタは、入力されたオブジェクトファイル中の関数を検索する(ステップ801)。関数があった場合は、そのオブジェクトファイルの冗長テキストコードポインタ情報格納領域から、ロードファイルに格納する該当関数のプロファイルポインタ情報を作成する(ステップ802)。オブジェクトをリンケージした結果のロードプログラムに未解決シンボルがなくなれば、処理を終了し、あれば未解決シンボルを含むオブジェクトファイルをライブラリファイルより検索する(ステップ804)。検索が成功したら、ステップ801からを繰り返す。

【0019】図9は、ロードプログラムの変換方式のフローチャートである。履歴情報採取対象関数シンボルを、プロファイルポインタ情報から検索する(ステップ901)。見つからなければ、終了し、見つかった場合、フラグ判定処理に進む(ステップ903)。フラグがオンであれば、履歴採取用に変換済みであるとして、変換処理をスキップし終了する(ステップ904)。フラグがオフであった場合は、プロファイルポインタ情報より該当関数の冗長コードを履歴採取関数に分岐するよう命令を変更する(ステップ905)

【0020】

【発明の効果】本発明では、実行履歴を採取するためにソースプログラムを再コンパイルすることなしに、本番運用用のロードプログラムの全部または特定の関数を選択して実行履歴採取用のロードプログラムに変換する手段と、逆に実行履歴採取用のロードプログラムの全部または特定の関数を実行履歴を採取しないように変換できるため、性能解析の効率化ができる。

【図面の簡単な説明】

【図1】本発明の一実施例のファイル構成を示す図である。

【図2】本発明の一実施例をファイル構成を示すフローチャートである。

【図3】オブジェクトファイルのテキスト命令展開方法を示す図である。

【図4】ロードプログラムのリンケージ方式を示す図である。

【図5】ロードプログラムのテキスト命令変換方式を示す図である。

【図6】ロードプログラムのファイル形式を示す図である。

【図7】プロファイルポインタ情報の詳細な形式を示す図である。

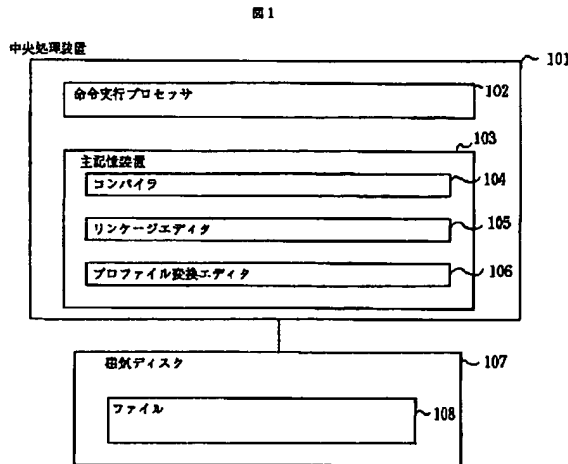
【図8】リンケージ方式を示すフローチャートである。

【図9】ロードプログラムの変換方式を示すフローチャートである。

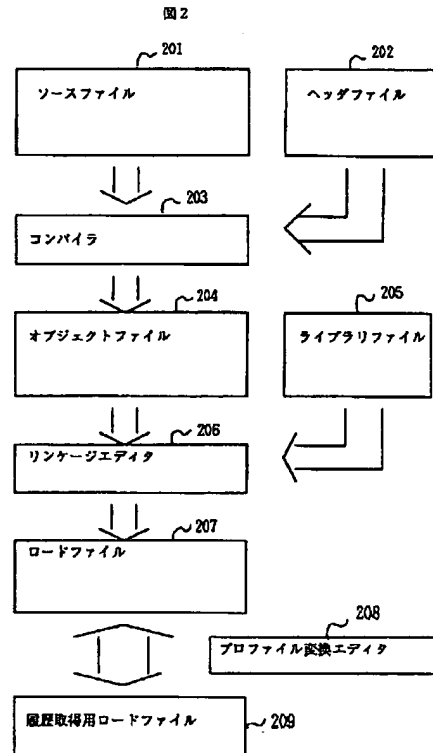
【符号の説明】

201…ソースファイル、203…コンパイラ、204…オブジェクトファイル、206…リン*
 *ケーリエディタ、207…ロードファイル、208…プロフィール変換エディタ、209…履歴取得用ロードファイル。

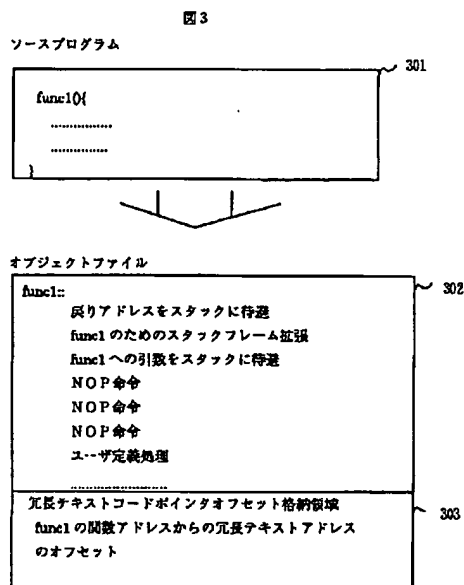
【図1】



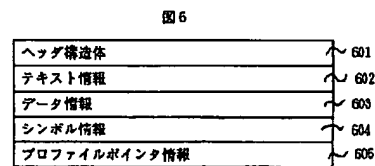
【図2】



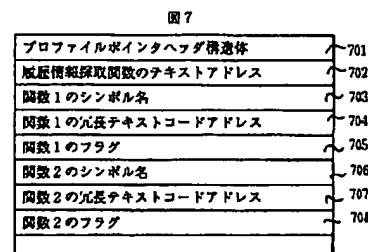
【図3】



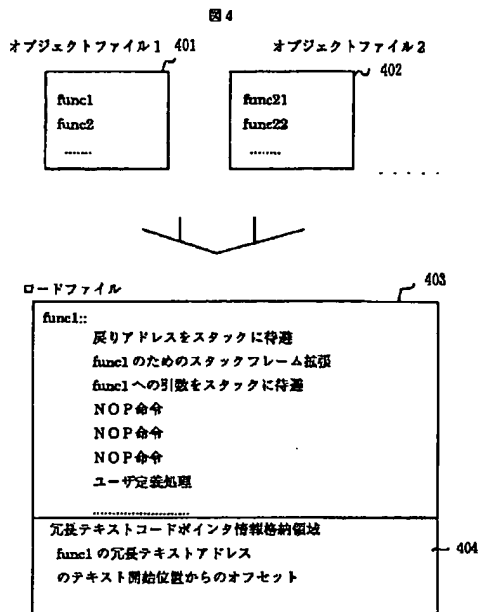
【図6】



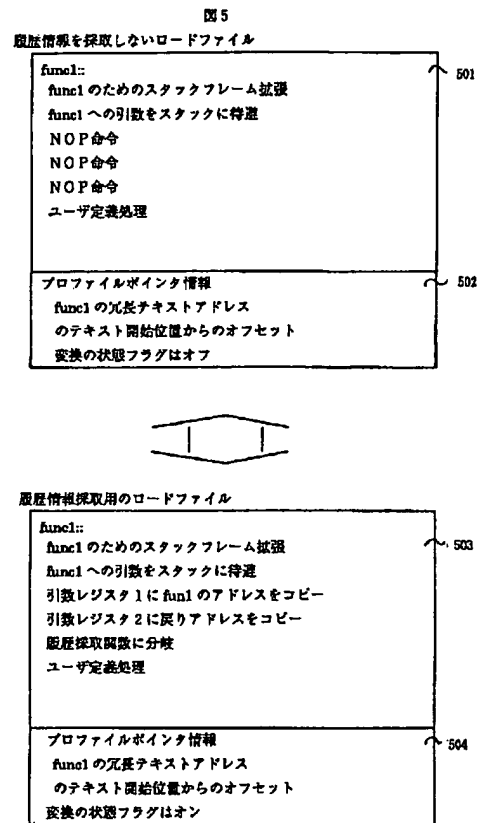
【図7】



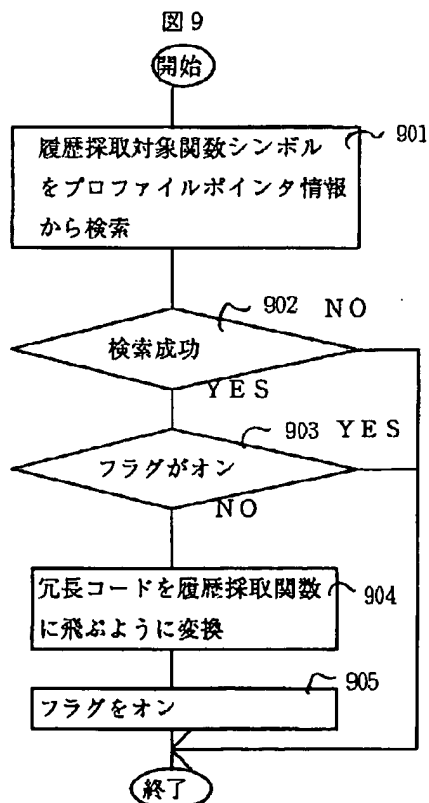
【図4】



【図5】

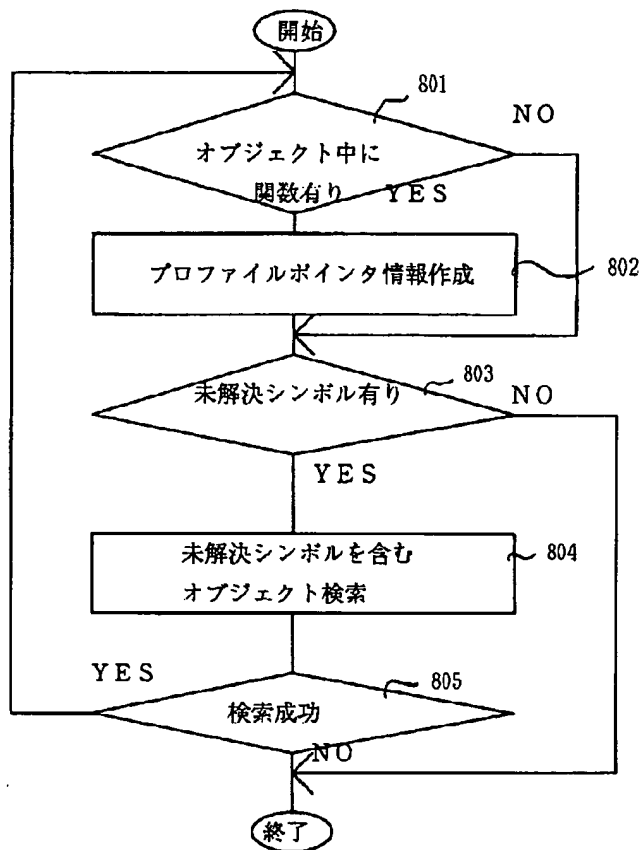


【図9】



【図8】

図8



フロントページの続き

(72)発明者 小原 勲

Fターム(参考) 5B042 FD02 FD13

神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア開発本部内